

Pengenalan Angka Tulisan Tangan Dengan Penerapan Freeman Chain Code yang Dimodifikasi

Handwritten Numeral Recognition by application of modified Freeman Chain Code

Fitri Amalia Hadi¹, Tjokorda Agung Budi W.², Kurniawan Nur Ramadhani³
¹²³Fakultas Informatika, Telkom University, Bandung fitriamalia13@gmail.com¹,
cokagung@telkomuniversity.ac.id², andiess2006@gmail.com³

Abstrak

Sistem pengenalan angka tulisan tangan dapat dimanfaatkan pada rekapitulasi perhitungan suara Pemilihan Presiden dan Wakil Presiden 2014 yang dicatat pada formulir model C1. Namun setiap orang memiliki cara penulisan angka yang berbeda, sehingga sistem pengenalan angka tulisan tangan dibangun diharapkan dapat mengenali angka dengan baik.

Sistem ini terdiri dari beberapa tahap, yaitu tahap *preprocessing*, ekstraksi ciri, dan klasifikasi. Ciri dari citra kerangka objek hasil *preprocessing* dibangun metode Freeman Chain Code yang dimodifikasi dengan melakukan pembagian region citra, pembentukan histogram chain code, dan penambahan 4 ciri visual. Selanjutnya hasil ekstraksi ciri diujikan menggunakan dua buah metode klasifikasi yaitu K-NN dan JST *Backpropagation*.

Hasil pengujian menunjukkan bahwa metode ekstraksi ciri Freeman Chain Code yang dimodifikasi mampu mengenali karakter angka tulisan tangan dari kedua data set dengan cukup baik. Dengan melakukan pembagian 9 region citra, normalisasi histogram chain code, dan penambahan 4 ciri, tingkat akurasi terbaik dapat mencapai 89.82% menggunakan K-NN dan mencapai 93.60% menggunakan JST *Backpropagation*.

Kata Kunci: *Offline*, pengenalan angka tulisan tangan, Freeman Chain Code

1. Pendahuluan

Pada tanggal 9 Juli 2014, Indonesia menyelenggarakan Pemilu Presiden dan Wakil Presiden (PPWP) 2014. Setiap warga negara dapat menggunakan haknya untuk memilih di TPS daerah yang telah ditentukan. Setiap TPS daerah menghitung dan mencatat hasil pemungutan suara pada formulir model C1. Formulir model C1 merupakan formulir yang digunakan dalam pencatatan data jumlah pemilih dan hasil pemungutan suara masing-masing pasangan calon presiden dan wakil presiden di tingkat TPS pada pelaksanaan PPWP 2014. Formulir C1 kemudian dikumpulkan dan rekapitulasi nasional dilakukan oleh petugas KPU RI yang berpusat di Jakarta. Rekapitulasi nasional menghabiskan waktu tiga hari, cukup lama karena hasil perhitungan suara se-Indonesia direkap secara manual oleh petugas KPU RI. Untuk membantu peran petugas KPU RI dalam perhitungan suara, maka dalam penelitian tugas akhir ini dibangun sistem yang dapat melakukan pengenalan dan interpretasi karakter angka tulisan tangan pada formulir model C1 menggunakan bantuan komputer. Sistem ini dikenal dengan sebutan *Optical Character Recognition (OCR)*.

OCR merupakan salah satu kemampuan komputer untuk mengenali dan menerjemahkan karakter pada sebuah dokumen fisik berupa gambar, foto, dan dokumen lainnya ke dalam teks komputer [4]. Aplikasi OCR berupa pengenalan karakter tulisan tangan, yang lebih sering disebut *Handwritten Character Recognition*, terkenal cukup rumit karena setiap individu memiliki karakter tulisan tangan yang berbeda satu sama lain. Namun dalam pemanfaatannya, pengenalan karakter tulisan tangan sangat membantu dalam memudahkan pekerjaan manusia. Oleh karena itu, pada tugas akhir ini sistem pengenalan angka tulisan tangan *offline* pada formulir model C1 dibangun agar proses rekapitulasi dapat diselesaikan dengan cepat.

Sistem pengenalan angka tulisan tangan dibangun dengan menerapkan metode ekstraksi ciri Freeman Chain Code yang diadopsi dari [17] dengan beberapa modifikasi. Modifikasi yang dilakukan antara lain menentukan titik mulai untuk membentuk histogram chain code berdasarkan pembagian region pada citra kerangka angka tulisan tangan, melakukan normalisasi nilai histogram chain code, dan menambahkan 4 ciri visual. Tiga ciri visual diambil dari penelitian [17]. Dua data set digunakan untuk menguji ciri yang dibentuk, yaitu data set *Mixed National Institute of Standards and Technology (MNIST)* dan data set formulir model C1. Kelayakan ciri yang dibentuk dari kedua data set diuji menggunakan dua metode klasifikasi, yaitu metode K-NN dan Jaringan Syaraf Tiruan (JST) *Backpropagation*. Hasil tingkat akurasi dari K-NN menunjukkan patokan baik buruknya atau kelayakan ciri yang dibentuk menggunakan Freeman Chain Code yang dimodifikasi. Sedangkan metode JST *Backpropagation* digunakan untuk mengamati seberapa besar tingkat keberhasilan yang dapat dicapai oleh sistem yang dibangun.

2. Optical Character Recognition (OCR)

Pengenalan karakter optis, atau lebih populer disebut OCR (*Optical Character Recognition*), merupakan salah satu kemampuan komputer untuk membaca dan mengenali karakter ke dalam teks komputer dari dokumen-dokumen cetak dalam bentuk citra digital. Karakter dapat berupa karakter cetak/latin maupun karakter tulisan tangan (*handwritten character*). Seiring dengan berkembangnya riset mengenai OCR, permintaan untuk mengenali karakter tulisan tangan pun bertambah akibat banyaknya data (seperti alamat yang ditulis pada amplop; nilai uang yang ditulis dalam nota; nama, alamat, nomor identitas, dan nilai *dollar* yang ditulis dalam faktur dan formulir) ditulis dengan tangan dan harus dimasukkan ke dalam komputer untuk diproses [3].

Pada OCR, karakter pada dokumen yang telah dipindai menggunakan *scanner* atau alat pindai optis lainnya akan melalui 3 tahap pada sistem pengenalan [3], yaitu sebagai berikut.

1. Tahap *preprocessing* untuk meningkatkan kualitas citra masukan dan menemukan karakter yang akan dikenali.
2. Tahap Ekstraksi Ciri untuk mengambil ciri pembeda dari citra karakter digital untuk pengenalan.
3. Tahap Klasifikasi untuk memproses vektor ciri yang mengidentifikasi citra karakter dan kata.

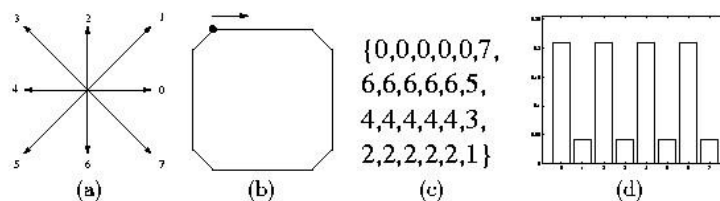
Angka tulisan tangan merupakan salah satu karakter cukup sulit untuk dikenali. Setiap orang memiliki cara penulisan angka yang berbeda, bahkan karakter angka yang ditulis oleh seseorang pun tidak selalu sama. Keberagaman tulisan tangan angka ini menjadi salah satu topik yang menantang dalam riset OCR.

3. Freeman Chain Code

Chain code atau kode rantai pertama kali dikenalkan pada tahun 1961 oleh Freeman, sehingga dikenal dengan nama Freeman Chain Code. Kode rantai merupakan salah satu cara yang sederhana untuk merepresentasikan sebuah citra. Pengambilan ciri dengan kode rantai memerlukan citra biner sebagai masukan (*input*). Cara kerja kode rantai adalah dengan merepresentasikan citra berdasarkan garis batasnya (kontur).

Kode rantai dihasilkan dengan cara mengikuti garis konturnya dengan mengikuti arah kontur menggunakan skema 4 arah mata angin (*4-directional*) dan 8 arah mata angin (*8-directional*). Kode ini mengikuti kontur searah jarum jam dan memantau arah kepergian dari satu kontur *pixel* ke *pixel* berikutnya. Metode ini secara umum tidak dapat diterima karena dua alasan utama: (1) kode rantai yang dihasilkan cenderung cukup panjang, dan (2) setiap gangguan kecil di sepanjang garis kontur akibat adanya *noise* pada citra atau segmentasi yang tidak sempurna menyebabkan perubahan dalam kode yang mungkin tidak berhubungan dengan bentuk dari garis kontur. Kode rantai dari sebuah garis kontur tergantung pada titik awalnya. Kode yang dihasilkan dapat dinormalisasi [5]. Salah satunya menggunakan histogram chain code.

Histogram Chain Code dimaksudkan untuk mengelompokkan objek-objek yang terlihat mirip dari pengamatan manusia. Hal ini tidak dimaksudkan untuk hasil deteksi dan klasifikasi yang tepat. Histogram Chain Code dihitung dari presentasi sebuah kontur. Contohnya ditunjukkan pada Gambar 3.2.



Gambar 3.1 (a) Skema 8 arah mata angin chain code ($K=8$), (b) objek sederhana berupa persegi empat, (c) presetasi chain code objek persegi empat, (d) histogram chain code persegi empat.

Frekuensi kemunculan arah pada suatu *link* chain code diubah menjadi bin-bin yang merepresentasikan arah chain code. Dari histogram chain code terlihat karakteristik sebuah pola karakter berdasarkan jumlah frekuensi kode arah pada tiap bin. Nilai frekuensi tiap bin histogram dapat dinormalisasi menjadi peluang kemunculan dalam satu chain code menggunakan persamaan (2.1) [7].

$$P_{\text{arah}} = \frac{f_{\text{arah}}}{N}, \quad \text{arah} = 0, 1, \dots, K-1. \quad (3.1)$$

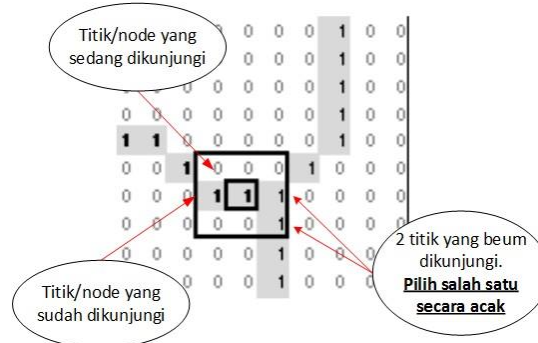
- P_{arah} : Probabilitas kemunculan sebuah kode arah chain code
- f_{arah} : Frekuensi kode arah chain code
- N : Jumlah kode arah pada sebuah chain code

Freeman Chain Code dipilih sebagai representasi karakter pada sebuah citra. Pada pengenalan karakter tulisan tangan, sering menemukan beberapa cabang dan hal ini membuat sulit untuk memutuskan harus pergi ke

arah mana. Selain itu, kembali ke titik yang sudah dikunjungi sebelumnya sering dibutuhkan untuk mengunjungi semua titik. Kesulitan-kesulitan ini menjadi alasan untuk menggunakan metode heuristik.

Metode heuristik sangat baik dalam mencari perkiraan solusi tetapi memiliki kompleksitas waktu dan ruang yang memungkinkan. Kompleksitas waktu mengacu pada waktu *running* program, dan kompleksitas ruang mengacu pada jumlah memori komputer. Karakteristik dari heuristik adalah [12]:

- 1) mudah diimplementasikan,
- 2) mudah di desain,
- 3) cepat meski tidak ada jaminan melebihi ekspektasi *run time*, dan
- 4) menghasilkan solusi yang baik meski tidak ada jaminan bahwa solusi sudah optimal atau mendekati optimal.



Gambar 3.2 Ilustrasi kondisi pemilihan titik tetangga secara acak

Pada tugas akhir ini, mengacu pada [12], metode heuristik yaitu algoritma acak (*randomized-based algorithm*) digunakan untuk melakukan penelusuran arah chain code dan pemilihan *starting nodes*/titik mulai pada chain code, ilustrasi ditunjukkan pada Gambar 3.2. Algoritma acak adalah algoritma yang membuat pilihan acak (*pseudo-random*). Algoritma acak yang bagus adalah yang mendapat hasil yang benar dengan peluang yang tinggi [13]. Algoritma acak dapat mengoptimalkan panjang chain code menjadi minimum.

4. Klasifikasi

Klasifikasi merupakan tahap akhir pada sistem pengenalan karakter angka tulisan tangan. Pada tahap ini, pelatihan dan pengujian dilakukan untuk mengetahui kualitas ciri dari hasil ekstraksi ciri yang telah didapat. Metode klasifikasi yang digunakan untuk pengujian metode ekstraksi ciri pada tugas akhir ini yaitu K-NN dan JST *Backpropagation*.

4.1 K-Nearest Neighbor (KNN)

Klasifikasi K-Nearest Neighbor (KNN) yang mendasar cukup mudah, sampel dikelompokkan berdasarkan kelas dari tetangga terdekatnya [4]. Ada banyak cara untuk menentukan kelas dari sebuah sampel, cara yang paling mudah adalah dengan memilih kelas yang paling banyak antara tetangga terdekatnya. Jadi KNN terdiri dari dua tahap; tahap pertama menentukan jumlah tetangga terdekat, tahap kedua menentukan kelas berdasarkan jumlah kelas yang telah ditentukan [4]. Cara lainnya adalah dengan menggunakan pengukuran metrik untuk mengukur kedekatan suatu sampel dengan tetangga terdekatnya. Salah satu cara yang sederhana adalah pengukuran metrik menggunakan Euclidean Distance [10].

Metode K-NN dipilih karena pada proses klasifikasi tidak ada bantuan elemen lain yang digunakan untuk menggolongkan sebuah data pada suatu kelas. Pengelompokan dilakukan hanya dengan mengukur jarak kedekatan dua data atau lebih berdasarkan nilai cirinya. Dalam implementasinya, pengenalan dengan K-NN dibantu menggunakan fungsi `knnclassify`.

4.2 Jaringan Syaraf Tiruan Backpropagation

Jaringan Syaraf Tiruan (JST) *Backpropagation* merupakan salah satu metode klasifikasi *supervised learning*. JST dilatih untuk dapat memetakan sekumpulan sampel *input-output* dengan akurasi yang tinggi [16]. Arsitektur JST *Backpropagation* yang digunakan terdiri dari 3 lapisan/layer, yaitu lapisan masukan (*input layer*), satu lapisan tersembunyi (*hidden layer*), dan lapisan keluaran (*output layer*). Nilai-nilai yang melewati setiap *neuron* pada jaringan akan diaktifkan oleh fungsi aktivasi. Fungsi aktivasi yang digunakan pada tugas akhir ini adalah fungsi sigmoid atau logistik.

Hidden layer berfungsi sebagai tempat untuk meng-*update* bobot yang digunakan untuk meminimalkan *error* sehingga nilai *output* pada jaringan dapat mendekati nilai target keluaran. Namun salah satu permasalahan

umum pada JST adalah menentukan jumlah *neuron* pada *hidden layer*. Rumus penentuan yang akan digunakan pada tugas akhir ini adalah rumus yang diusulkan Shibata dan Ikeda yang dikutip pada [15] sebagai berikut.

$$H = \sqrt{IN} \quad (4.1)$$

H : Jumlah *hidden neuron*
 I : Jumlah *input neuron*
 N : Jumlah *output neuron*

Dalam pembelajaran, JST *Backpropagation* melalui dua fase utama, yaitu fase propagasi maju dan fase propagasi mundur. Pada implementasinya dibantu menggunakan *pattern recognition tool* pada MATLAB.

5. Rancangan Sistem

Proses utama dalam sistem terdiri dari 3 tahap yaitu *preprocessing* citra, ekstraksi ciri menggunakan metode Freeman Chain Code yang dimodifikasi, dan klasifikasi menggunakan metode K-NN dan JST. Tahap *preprocessing* pada data set MNIST dan data set form C1 dibedakan. Tujuan dari kedua *preprocessing* sama yaitu membentuk citra biner *thinning*, atau dapat kita sebut *Thinned Binary Image* (TBI). Hasil *preprocessing* citra MNIST ditunjukkan pada Gambar 5.1 dan hasil *preprocessing* citra form C1 ditunjukkan pada Gambar 5.2.



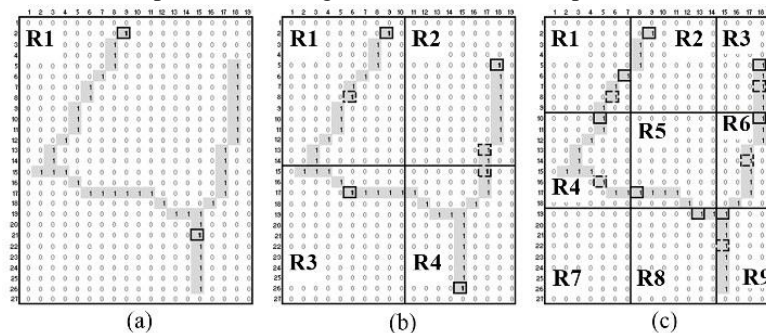
Gambar 5.1 Hasil *preprocessing* MNIST asli hingga citra *thinning*



Gambar 5.2 Hasil *preprocessing* form C1 asli hingga *thinning*

Citra hasil *thinning* (TBI) digunakan pada proses pembentukan ciri dengan metode Freeman Chain Code yang dimodifikasi. Penelusuran titik menggunakan algoritma acak digunakan pada pembentukan histogram chain code. Tahap proses ekstraksi ciri adalah sebagai berikut.

1. Pembagian region dan pemilihan titik mulai dari region yang telah dibagi, pilih 2 titik mulai dari masing-masing region. Satu titik dipilih dari sudut gambar dan satu titik dipilih acak.



Gambar 5.3 Ilustrasi pembagian region pada citra *thinning*

Detail jumlah ciri tiap region ditampilkan pada Tabel 5.1.

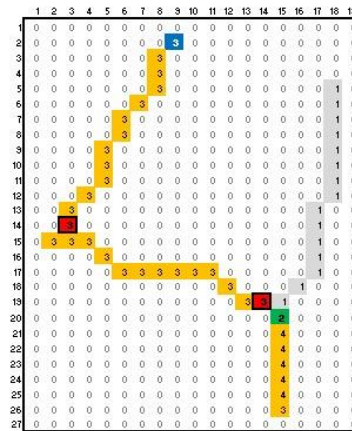
Tabel 5.1 Panjang vektor ciri berdasarkan pembagian region

No.	Region	Jumlah titik per region	Jumlah ciri chain code	Jumlah ciri tambahan	Panjang vektor ciri
1	1	2	$1 \times 2 \times 8 = 16$	4	20
2	4	2	$4 \times 2 \times 8 = 64$	4	68
3	9	2	$9 \times 2 \times 8 = 144$	4	148

Pada tahap selanjutnya, diambil contoh citra yang dibagi sebanyak 1 region dengan jumlah 20 ciri.

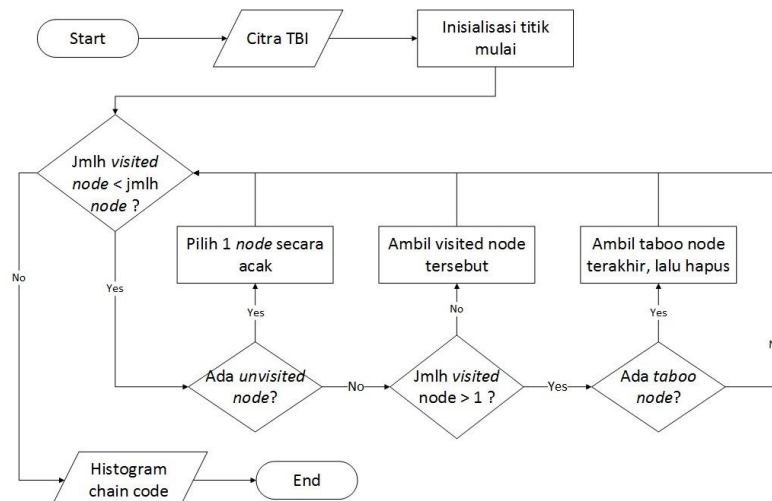
2. Dari titik-titik mulai yang didapat, secara berurutan (dari region pertama hingga terakhir) telusuri arah chain code menggunakan algoritma acak, lalu simpan rangkaian chain code tersebut ke dalam histogram chain code yang digabung menjadi sebuah vektor ciri chain code. Pada saat penelusuran rute arah sedang

berlangsung akan ditemui tiga jenis *node*, yaitu *node* yang belum dikunjungi (*unvisited node*), *node* yang sudah dikunjungi (*visited node*), dan *node* yang terakhir dikunjungi pada percabangan atau *node* yang memiliki dua *node* tetangga atau lebih yang belum dikunjungi (*taboo node*). Contoh jenis titik/*node* ditunjukkan pada Gambar 5.4.



Gambar 5.4 Citra *thinning* (TBI) dengan simbol-simbol *node*

Sebagai label, *unvisited node* ditandai dengan nilai 1, *visited node* ditandai dengan nilai 3, dan *current node* (*node* yang sedang dikunjungi) ditandai dengan nilai 2. Untuk keperluan *revisit*, maka *revisit node* ditandai dengan nilai 4. *Node-node* yang berpotensi sebagai *node taboo* adalah *node* yang ditandai dengan warna merah dan kotak tebal. Penelusuran titik ditunjukkan pada Gambar 5.5.



Gambar 5.5 Diagram alir algoritma acak dalam penelusuran titik

3. Lakukan normalisasi nilai frekuensi histogram chain code.
4. Gabungkan vektor ciri histogram chain code dengan vektor 4 ciri tambahan sebagai berikut.
 - a. Rasio-atas (r_a), perbandingan jumlah titik/*node* di area bagian atas pada citra yang telah dibagi 2 secara vertikal dengan jumlah *node* pada citra.

$$r_a = \frac{\text{Jumlah node bagian atas}}{\text{Jumlah node total}} \quad (3.2)$$

- b. Rasio-kanan (r_k), perbandingan jumlah *node* di area bagian kanan citra yang telah dibagi 2 secara horizontal dengan jumlah *node* pada citra.

$$r_k = \frac{\text{Jumlah node bagian kanan}}{\text{Jumlah node total}} \quad (3.3)$$

- c. Rasio-luas (r_l), perbandingan luas karakter murni (tanpa ruang kosong/*padding*) dengan luas citra maksimum yang ada di data set (28×28 *pixel*). Luas karakter murni pada citra TBI angka “4” yaitu 25×17 *pixel*.

$$z = \frac{\begin{matrix} \text{0000} & \text{00000000} \\ \text{0000} & \text{00000000} \end{matrix}}{\begin{matrix} \text{0000} & \text{00000000} & \text{0000} & \text{0000} \end{matrix}} \quad (3.4)$$

- d. Rasio-tinggi (r_t), perbandingan tinggi karakter dengan tinggi citra karakter.

$$r_t = \frac{\begin{matrix} \text{00000} & \text{00000000} \end{matrix}}{\begin{matrix} \text{00000} & \text{00000} & \text{00000} \end{matrix}} \quad (3.5)$$

Contoh hasil vektor ciri akhir yang terbentuk dari citra dengan pembagian 1 region ditampilkan pada Tabel 5.2. Kumpulan vektor ciri dari semua data set digunakan sebagai *input* pada tahap klasifikasi.

Tabel 5.2 Hasil vektor ciri akhir pada pembagian citra 1 region

d11	...	d18	d21	...	d28	f1	f2	f3	f4
0.17021	...	0.1064	0.02128	...	0	0.48	0.5625	0.542	0.926

6. Pengujian

Pengujian sistem dilakukan untuk menganalisis kelebihan dan kekurangan sistem pengenalan angka tulisan tangan yang dibangun menggunakan metode ekstraksi ciri Freeman Chain Code yang dimodifikasi. Implementasi dan pengujian sistem pengenalan angka tulisan tangan menggunakan perangkat lunak MATLAB R2013a.

6.1 Data Uji

Data set yang digunakan ditunjukkan pada Tabel 6.1.

Tabel 6.1 Data Uji

Data set	Jumlah Sampel <i>Training</i>	Jumlah Sampel <i>Testing</i>
MNIST	50000	10000
Form C1	2100	550

6.2 Skenario Pengujian

Pengujian sistem pengenalan angka tulisan tangan dilakukan dengan 3 skenario pengujian berikut ini.

1. Skenario 1 – Pembagian region dan normalisasi histogram chain code

Pengujian dilakukan terhadap ciri citra data set MNIST berdasarkan pembagian citra (1 region, 4 region, dan 9 region) dan normalisasi nilai frekuensi histogram chain code (HCC). Hasil pengujian dibandingkan untuk melihat pengaruh pembagian region dan normalisasi HCC terhadap akurasi pengenalan angka tulisan tangan. Parameter terbaik yang didapat selanjutnya digunakan pada pengujian skenario 2 dan 3.

2. Skenario 2 – Pengujian MNIST-C1

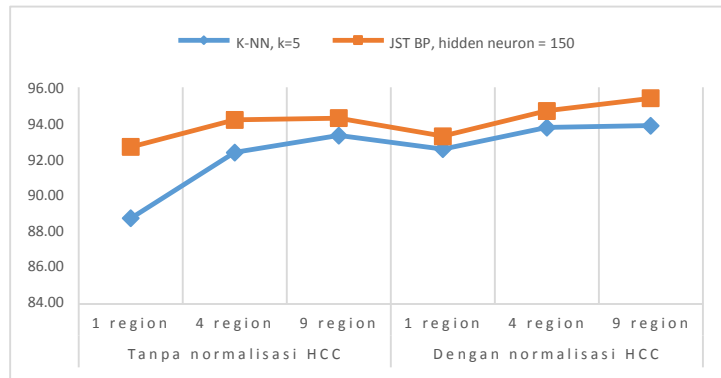
Skenario kedua, hasil pengenalan/pengujian menggunakan data *testing* ciri form C1 dianalisis berdasarkan data *training* ciri MNIST menggunakan parameter ciri terbaik.

3. Skenario 3 – Pengujian C1-C1

Skenario terakhir menggunakan data set form C1 murni, artinya data pelatihan menggunakan data *training* ciri form C1 dan data uji menggunakan data *testing* ciri form C1. Data *testing* form C1 sebanyak 550 sampel diuji berdasarkan pelatihan data *training* form C1 sebanyak 2100 sampel. Selain itu, pengaruh 4 ciri tambahan yaitu rasio-atas, rasio-kanan, rasio-luas, dan rasio-tinggi dianalisis bila dihilangkan dari vektor ciri.

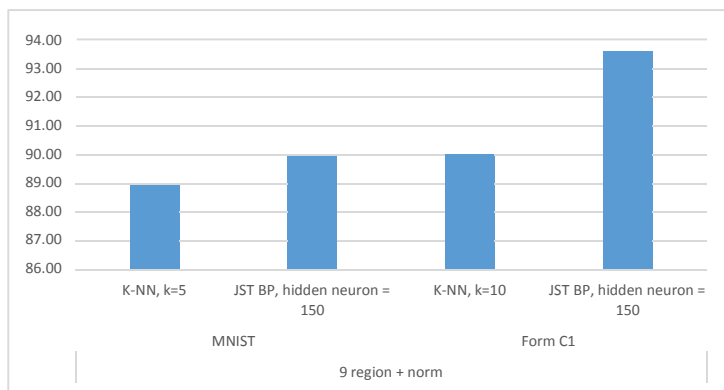
7. Hasil Pengujian

Pengujian skenario pertama membandingkan ciri dengan tiga pembagian region yang berbeda pada data *testing* MNIST terhadap data *training* MNIST (MNIST-MNIST), yaitu ciri 1 region, ciri 4 region, ciri 9 region. Penggunaan normalisasi nilai frekuensi pada histogram chain code (HCC) juga dibandingkan. Tingkat akurasi pengujian dengan metode K-NN dengan $k = 5$ dan JST *Backpropagation* (JST BP) dengan hidden neuron 150 dapat dilihat pada Gambar 7.1. Berdasarkan hasil pengujian, dapat disimpulkan bahwa pembagian citra sebanyak 9 region mampu mengenali karakter angka tulisan tangan pada data *testing* MNIST dengan akurasi yang cukup tinggi, yaitu dengan akurasi di atas 88%. Hal ini menunjukkan ciri yang dihasilkan menggunakan metode Freeman Chain Code yang dimodifikasi sudah baik untuk mengenali karakter angka tulisan tangan pada data set MNIST.



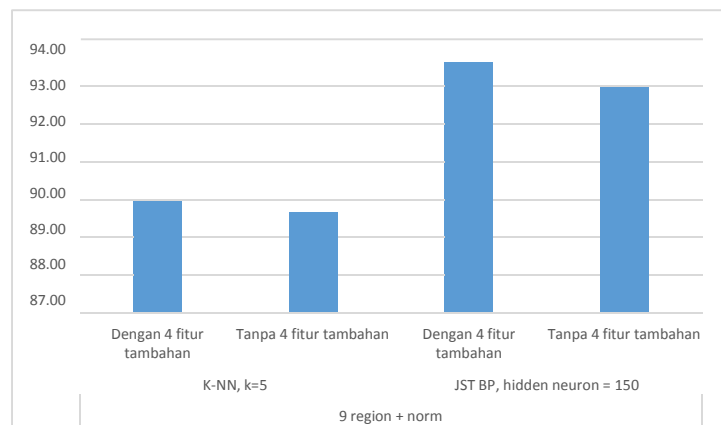
Gambar 7.1 Grafik tingkat akurasi MNIST-MNIST berdasarkan Pembaian Citra & Normalisasi HCC

Seperti yang ditunjukkan Gambar 7.1, semakin banyak pembagian region tingkat akurasi meningkat. Ini berarti vektor ciri yang dihasilkan dapat menunjukkan keunikan dalam merepresentasikan setiap karakter angka tulisan tangan pada kelas angka sebenarnya, karena pola karakter dilihat dari berbagai titik yang berbeda. Pengaruh penggunaan normalisasi nilai frekuensi histogram chain code pun didapat mampu meningkatkan akurasi pengenalan.



Gambar 7.2 Grafik tingkat akurasi pengujian data *testing* form C1 berdasarkan data *training* MNIST dan data *training* form C1

Hasil pengujian skenario 2 pada Gambar 7.2 menunjukkan proses pengolahan citra/*preprocessing* pada kedua data set sangat berpengaruh pada citra yang dihasilkan. Tingkat akurasi yang cukup baik menunjukkan rangkaian *preprocessing* form C1 yang berbeda dari MNIST dapat merepresentasikan pola-pola karakter angka tulisan tangan dengan baik. Penggunaan data *training* pun berpengaruh pada hasil pengujian.



Gambar 7.3 Grafik tingkat akurasi pengaruh penambahan 4 ciri visual pada pengujian menggunakan data *testing* form C1 berdasarkan data *training* form C1

Berdasarkan Gambar 7.3, tingkat akurasi yang didapat dari pengujian ciri jika hanya menggunakan 144 ciri (tanpa menggunakan 4 ciri tambahan) pada pengujian dengan K-NN menurun 0.18% dari 89.92% dan pada

pengujian dengan JST BP menurun 0.7% dari 93.6%. Dari pengujian ini dapat disimpulkan bahwa penggunaan ciri histogram chain code dengan tambahan ciri visual berupa rasio-rasio dari sebuah citra karakter angka tulisan tangan mampu membuat akurasi pengenalan meningkat karena dengan menambahkan rasio-rasio kerangka pola angka dapat memperbesar perbedaan ciri pola angka dari suatu kelas dengan ciri pola dari kelas angka yang lainnya dan memperkecil perbedaan ciri pola angka di dalam kelas angka itu sendiri.

8. Kesimpulan

Dari beberapa pengujian dan analisis yang dilakukan, dapat diambil kesimpulan bahwa vektor ciri hasil ekstraksi ciri dengan metode Freeman Chain Code yang dimodifikasi menggunakan parameter pembagian 9 region dan normalisasi histogram chain code menghasilkan akurasi pengenalan terbaik. Citra pada data *testing* form C1 dapat dikenali dengan baik berdasarkan pelatihan pada data *training* MNIST dengan tingkat akurasi mencapai 88.91% menggunakan K-NN dan mencapai 90% menggunakan Jaringan Syaraf Tiruan *Backpropagation*. Namun tingkat akurasi pengenalan meningkat rata-rata sebesar 2.43% ketika data *testing* form C1 diuji berdasarkan data *training* form C1. Dengan menambahkan 4 ciri visual berupa rasio-atas, rasio-kanan, rasio-luas, dan rasio-tinggi kerangka angka pada citra, tingkat akurasi pengenalan dapat meningkat dengan rata-rata sebesar 0.44%. Ciri yang dibangun menggunakan metode Freeman Chain Code yang dimodifikasi mampu mengenali citra karakter angka tulisan tangan pada pengujian menggunakan data *testing* form C1 berdasarkan pelatihan pada data *training* form C1 dengan akurasi tertinggi mencapai 89.82% menggunakan K-NN dengan $k=5$ dan mencapai 93.6% menggunakan Jaringan Syaraf tiruan *Backpropagation* dengan jumlah *hidden neuron*=150.

9. Daftar Pustaka

- [1] Anindita, R. 2012. *Pengenalan Huruf Alfabet Tulisan Tangan Menggunakan Modified Direction Feature dan Jaringan Syaraf Tiruan Backpropagation*. Tugas Akhir, Institut Teknologi Telkom, Bandung.
- [2] Azmi, N.N., Nasien, D. 2014. *Feature Vector of Binary Image Using Freeman Chain Code (FCC) Representation based on Structural Classifier*. Int. J. Advance Doft Comput. Appl., vol.6, No.2.
- [3] Burger, W. 2008. *Digital Image Processing: An Algorithmic Introduction using Java*. Springer.
- [4] Cheriet, M., dkk. 2007. *Character Recognition System, A Guide for Students and Practitioners*. Wiley-Interscience.
- [5] Choudary, A. dkk. 2011. *Handwritten Numeral Recognition Using Modified BP ANN Structure*. Springer.
- [6] Cunningham, P., dan Sarah J. D. 2007. *k-Nearest Neighbor Classifier*. Multiple Classifier System: 1-17.
- [7] Ebrahimzadeh, R., Jampour, M. 2014. *Efficient Handwritten Digit Recognition based on Histogram of Oriented Gradient and SVM*. International Journal of Computer Applications, vol. 104 no. 9, pp.10-13.
- [8] Gonzalez, R.C., Woods, R.E 2002. *Digital Image Processing Second Edition*. Prentice Hall Inc.
- [9] HIPR2. *Table of Content*. [Online] tersedia di: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/index.htm> [diakses pada Juni 2015]
- [10] Iamsa-at, S., Hurata, P. 2013. *Handwritten Character Recognition using Histogram of Oriented Gradient Features in Deep Learning of Artificial Neural Network*. International Conference on IT Convergence and Security (ICITCS), pp.1-5.
- [11] Iivarinen, J., dkk. 1997. *Comparison Of Combined Shape Descriptors For Irregular Objects*. Proc. 8th British Machine Vision Conference. Vol 2: 430-439.
- [12] Jain, A.K. 1989. *Fundamentals of Digital Image Processing*. Prentice Hall.
- [13] Kanchi. 2003. *Connect two pixels*. [Online] tersedia di: <http://www.mathworks.com/matlabcentral/fileexchange/4211-connect-two-pixels> [diakses pada Mei 2015].
- [14] Khamar, K. 2013. *Short Text Classification Using kNN Based on Distance Function*. IJARCCCE. Vol 2(4).
- [15] Lecun, Y., dkk. 1998. *Gradient-Based Learning Applied to Document Recognition*. *Proceedings of IEEE*, 86(11):2278-2324.
- [16] MATLAB. 2015. Mathworks Documentation. [Online] tersedia di: <http://www.mathworks.com/help/> [diakses pada April 2015]
- [17] Nasien, D., dkk. 2010. *Recognition of Isolated Handwritten Latin Character using One Continuous Route of Freeman Chain Code Representation and Feedforward Neural Network Classifier*. World Academy of Science, Engineering and Technology. Vol 4.
- [18] Nasien, D., dkk. 2011. *The Heuristic Extraction Algorithms for Freeman Chain Code of Handwritten Character*. International Journal of Experimental Algorithms. Vol 1(1): 1-20.

- [19] Prasetyo, E. 2011. *Pengolahan Citra Digital dan Aplikasinya menggunakan MATLAB*. Yogyakarta: Penerbit ANDI.
- [20] Sheela, K. G., dan S. N. Deepa. 2013. *Review on methods to fix number of hidden neurons in neural networks*. Mathematical Problems in Engineering.
- [21] Suyanto. 2007. *Artificial Intelligence: Searching, Reasoning, Planning, and Learning*. Bandung: Informatika.
- [22] Tiwari, R.R. dkk. 2013. *Handwritten Digit Recognition Using Backpropagation Neural Network & K-Nearest Neighbour Classifier*. International Journal of Electrical, Electronics and Data Communication, vol. 1.